

## Operating Systems Memory Management Assignment

Your next assignment is to create two memory management classes that swap processes into and out of memory using a **first fit** strategy and a **best fit** strategy. You must use a **linked list** representation to keep track of the contents of memory. You can use the `java.util.LinkedList` class if you like.

To complete this assignment you will be creating two Swapper classes that plug-into my memory management framework. Your Swapper classes should implement the following interface:

```
public interface ISwapper
{
    public void init(int memSize);
    public boolean load(IProcess p, IMemory mem) throws
        BlueScreenException;
    public void unload(IProcess p, IMemory mem) throws
        BlueScreenException;
}
```

The **init** method will be called by my framework after your Swapper is created. The init method is passed the total size of physical memory.

The **load** method will be called whenever your swapper class should load a process into memory. The load method is passed the process you should load and the physical memory object that the process should be loaded into. If there is room in memory you should load the process using the **first fit** strategy for your first Swapper class, and **best fit** for your second Swapper class. If there is no room in physical memory you should return false, otherwise return true.

The **unload** method will be called whenever a process should be swapped out of memory. The unload method is passed the process you should unload and the physical memory object that it should be unloaded from.

Your load and unload methods will use the IMemory object in order to load and unload processes to and from physical memory. Important methods of the IMemory object are shown below:

```
public interface IMemory
{
    public void load(IProcess p, int start, int end)
        throws BlueScreenException;
    public void unload(IProcess p) throws
        BlueScreenException;
    public int getSize();
    public void dump();
}
```

You should use the load and unload methods of the IMemory object to load a process into a location in memory or to unload a process. You can also find out how much physical memory exists by calling getSize. If you try to load a process into a memory location that is already occupied, or if you try to unload a process that is not currently in memory, an exception will be thrown. Finally, you can get a printout of the contents of memory by calling dump.

Your swapper classes will also need to interact with the IProcess object in order to get information about the process it is supposed to load or unload. Important methods of the IProcess object are shown below:

```
public interface IProcess
{
    public int getSize();
    public int getId();
}
```

The processes that must be loaded, and how long they will remain in memory, is defined by an input file. An example input file is provided below:

```
Duration in clock ticks, Memory required for process
3, 256
20, 512
3, 64
3, 64
20, 128
5, 32
10, 256
```

The first line in the file is ignored by the program and acts as a comment about what each item in the file represents. Each line after the first lists a process that needs to be loaded in memory. For example, the file above specifies that 7 processes need to be loaded into memory. The number of clock ticks that each process will remain in memory and the number of bytes that the process needs to run are given on each line in the file.

To test your program you can run the Main program in my framework (mem.jar) and pass it two parameters. The first parameter is the name of your Swapper class. The second parameter is the name of the input file that defines how the processes will be loaded. For example, assuming you're your first class is called MyFirstFitAlg, and the process information is in a file called input.txt, you could run the main program as follows:

```
java -classpath mem.jar Main MyFirstFitAlg input.txt
```

This assumes that the mem.jar file is in the current directory as well as the MyFirstFitAlg.class file.

I will be testing your classes with a few different input files. When you submit your assignment please include the source code for both of your Swapper classes